**Introduction**

Let’s take a quick look at a very important and common experimental problem: checking if the difference in success rates of two [Binomial experiments](https://en.wikipedia.org/wiki/Binomial_distribution) is statistically significant. This can arise in A/B testing situations such as online advertising, sales, and manufacturing.

We already share a [free video course on a Bayesian treatment of planning and evaluating A/B tests](http://www.win-vector.com/blog/2016/02/free-video-course-applied-bayesian-ab-testing-in-r/) (including a [free Shiny application](https://github.com/WinVector/CampaignPlanner_v3)). Let’s now take a look at the should be simple task of simply building a summary statistic that includes a classic frequentist significance.

Suppose our data comes from collecting outcomes from two processes: A and B, and is as follows.

# build example data

set.seed(2018)

A <- rep(0, 2000)

A[[sample.int](http://sample.int)(length(A), 823)] <- 1

B <- rep(0, 100)

B[[sample.int](http://sample.int)(length(B), 55)] <- 1

# summarize data

(kA <- sum(A))

## [1] 823

(nA <- length(A))

## [1] 2000

(kB <- sum(B))

## [1] 55

(nB <- length(B))

## [1] 100

(abs\_rate\_difference <- abs(nB\*kA - nA\*kB)/(nA\*nB))

## [1] 0.1385

df <- data.frame(

outcome = c(A, B),

treatment = c(rep("A", length(A)),

rep("B", length(B))),

stringsAsFactors = FALSE)

table(df)

## treatment

## outcome A B

## 0 1177 45

## 1 823 55

A natural question is: how likely is such a large difference of success rates assuming a null-hypothesis that both processes are in fact the same and generating 1’s at the common observed rate? This is a classic frequentist significance question.

**The sigr philosophy**

The philosophy of the the [sigr](https://cran.r-project.org/package=sigr) [R](https://www.r-project.org/) package is that each common statistical test should be a "one liner" that properly performs and formats a test. That is not to say statistical tests are easy to understand, they can in fact be difficult. It is the intent that performing and reporting the test should not *add* to the difficulty of finding, applying, and reporting the correct test.

So with correct documentation, training, and thought one can (with a reasonable effort) find, understand, and apply a correct test in a manner that *appears* effortless a-posteriori.

In the case of comparing rates of two Binomial experiments (the correct formulation of comparing rates in an A/B test) we say our mental handbook now includes sigr::Bernoulli\_diff\_stat() as the correct test.

**The sigr::Bernoulli\_diff\_stat() solution**

With the [sigr](https://cran.r-project.org/package=sigr) [R](https://www.r-project.org/) package we can answer the question directly with the difference in [Bernoulli processes difference statistic](https://winvector.github.io/sigr/reference/Bernoulli_diff_stat.html). Frequentest summaries are supposed to be easy and quick to derive, and here is the difference in rates of Binomial processes as a one-liner.

library("sigr")

s <- Bernoulli\_diff\_stat(kA, nA, kB, nB)

print(s)

## [1] "Bernoulli difference test: (A=823/2000=0.4115, B=55/100=0.55, post 0.1385 two sided; p=0.006062)."

The above is deliberately concise. It assumes we (and our partners) know exactly the test we want and quickly applies and formats (without explanation). We emphasize critical thought and explanation are critical in statistical testing- they are just not part of the test. We also say a longer test procedure (say 5, 10, or 20 lines of code) steals attention from thinking *about* the experiment. What we suggest is having concise and correct testing code ready, and leaving more time for considering (and confirming) we have in fact applied an appropriate test.

In addition to a formatted presentation the Bernoulli\_diff\_stat() returns a number of useful values.

s$probi # assumed common rate

## [1] 0.4180952

s$test\_rate\_difference # difference in rates we are testing for

## [1] 0.1385

s$pValue # estimated p-value/significance

## [1] 0.006062159

The small p-value indicates that a difference of this magnitude (0.1385) is unlikely under the null-assumption that both processes were in fact identical with a common success-rate (rate of 1‘s in each output) of 0.418095.

Bernoulli\_diff\_stat() is a deterministic and exact statistical test when max(nA, nB) %% min(nA, nB) == 0. It has minimal assumptions/approximations as it is directly testing a difference in scaled Binomial/Bernoulli processes. When max(nA, nB) %% min(nA, nB) != 0 the test gives a range of answers:

Bernoulli\_diff\_stat(824, 2001, 55, 100)

## [1] "Bernoulli difference test: (A~824/2001=0.4118, B=55/100=0.55, post 0.1382 two sided; pL=0.005978, pH=0.006067)."

The range is formed by alternately truncating and extending the longer process to get the required divisibility property. The reported range is tight when max(nA, nB) / min(nA, nB) is large. Being a statistic of discrete summaries Bernoulli\_diff\_stat() is sensitive to end-effects (the p-value moves up and down as exact counts move in and out of rate-defined intervals).

**Simulation estimate**

One could try to estimate the significance in difference in rates directly through re-sampling or simulation techniques. Obviously estimating small significances is going to be computationally expensive.

set.seed(2018)

n\_runs <- 100

run\_size <- 10000

mk\_resample <- function(A, B, run\_size) {

force(A)

force(B)

kA <- sum(A)

nA <- length(A)

kB <- sum(B)

nB <- length(B)

univ <- c(A, B)

nU <- length(univ)

force(run\_size)

# integer to avoid rounding issues

abs\_rate\_difference\_times\_nab <- abs(nB\*kA - nA\*kB)

function(...) {

nge <- 0

for(i in seq\_len(run\_size)) {

sA <- univ[[sample.int](http://sample.int" \t "_blank)(nU, nA, replace = TRUE)]

sB <- univ[[sample.int](http://sample.int" \t "_blank)(nU, nB, replace = TRUE)]

obs <- abs(nB\*sum(sA) - nA\*sum(sB))

nge <- nge + (obs>=abs\_rate\_difference\_times\_nab)

}

nge

}

}

f <- mk\_resample(A, B, run\_size)

cl <- parallel::makeCluster(parallel::detectCores())

res <- as.numeric(parallel::parLapply(cl, seq\_len(n\_runs), f))

parallel::stopCluster(cl)

k <- sum(res)

n <- n\_runs\*run\_size

print(k/n) # empiricl estimate of p-value/

## [1] 0.006011

We can, with a quick appeal to another canned test (wrapBinomTestS()), check if our original theoretical p-value (s$pValue = 0.00606216) is compatible with the empirical estimate of the same.

wrapBinomTestS(k, n, p = s$pValue)

## [1] "Exact binomial test: (6011/1e+06=0.006011~c(0.95)[0.00586, 0.006164], two.sided 0.006062; p=n.s.)."

The "p=n.s." means the Bernoulli\_diff\_stat() value is close the the empirical value (as we want).

**Other estimates**

A common (and useful) way to estimate the difference in Bernoulli/Binomial rates is to leave out some domain knowledge to convert to a nearly equivalent problem that itself has a ready-made test. Two common such relaxations are forgetting that the process is Bernoulli/Binomial 0/1 and using general t-test to compare means, or switching from a test of differences of rates and appealing to a Fischer Test of independence between process label and outcome marks (a related but different question).

**t-test estimate**

If we ignore the fact that the Binomial process generates only 0/1 we can use a classic t-test to estimate the significance of the observed difference.

ttest <- wrapTTest(A, B)

print(ttest)

## [1] "Welch Two Sample t-test, two.sided: (t=-2.705, df=108.8, p=0.007925)."

Notice the t-test is not in the confidence interval of empirical estimates of the differences in rates significance.

wrapBinomTestS(k, n, p = ttest$tt$p.value)

## [1] "Exact binomial test: (6011/1e+06=0.006011~c(0.95)[0.00586, 0.006164], two.sided 0.007925; p<1e-05)."

One could work further on the above by adding "continuity corrections" (which is really just a vainglorious way of say "shifting boundaries by 0.5") and so on, but we feel running an exact test that matches the problem’s actual generative model (and actual distribution) is preferred.

**Fisher test estimate**

Or, if we ignore distinction between treatment and outcome we can use a Fisher independence test.

tab\_test <- wrapFisherTest(df, "treatment", "outcome")

print(tab\_test)

## [1] "Fisher's Exact Test for Count Data: (odds.ratio=1.747, p=0.00685)."

wrapBinomTestS(k, n, p = tab\_test$ft$p.value)

## [1] "Exact binomial test: (6011/1e+06=0.006011~c(0.95)[0.00586, 0.006164], two.sided 0.00685; p<1e-05)."

Again, the Fisher test is not in the confidence interval of empirical estimates of the differences in rates significance.

**Conclusion**

All of the above estimation methods are standard and serviceable (give the same qualitative answer that the difference in rates is in fact significant).

However notice, neither of the t-test estimate or the Fisher test estimate were in the empirically estimated confidence interval of the exact quantitative value for the unknown significance.

Because Bernoulli\_diff\_stat() has a low degree of approximations and assumptions we feel it is easy to use and explain (where it applies, in particular when max(nA, nB) %% min(nA, nB) == 0 or when max(nA, nB) / min(nA, nB) is large).

Bernoulli\_diff\_stat() is new code (so we are still exploring corner cases), but is based on sound concepts (converting differences in rates into differences in counts, an idea inspired by Wald’s *Sequential Analysis*).

The above are all examples of the sigr package philosophy: correct and concise wrappers for important statistical tests. Link to the correct statistics found in R but try to stay out of the user’s way both in performing the test and reporting the test.